Unit Summary Good Example

Note: This is only meant to be an example of unit summary content. The length is going to be different than yours because this is only one topic and you have multiple topics to cover.

Task 1:

LinkedLists:

- LinkedLists are a data structure that fulfill the List interface. Their underlying structure contains nodes that reference other nodes, "linking" together to form a list
    - This is in contrast to an ArrayList, which stores elements in a contiguous array
- In the implementation we used, each node only had a pointer to the next node, and the list only had a pointer to the front.
    - This is really inefficient since it means you have to traverse the whole list one by one for many common functions, resulting in the following runtimes:
        - Contains(value) - O(n)
        - Get(index) - O(n)
        - Set(index, value) - O(n)
        - remove(value) - O(n)
- A better implementation would have nodes that store references to both the next and the previous nodes, and the list would keep track of both the back and the front.
    - Though some of our big-O complexities wouldn't change, this would allow us to traverse both directions and make operations at both the front and the back super quick.
- Because LinkedLists have quick operations at the front and the back of the list, they are often used to implement Stacks and Queues.
    - Unlike in an ArrayList, where you would have to shift when adding/removing from the front

Task 2:

Studying LinkedLists helped me learn more about the difference between an interface (List) and an implementation (ArrayList vs. LinkedList), which will help me better understand the best way to organize data structures in my programs. I also gained valuable experience working with pointers and reference semantics, both of which are commonly used in many programming situations. Another thing I learned more about was how to consider the tradeoffs between different data structures, for example comparing the runtime of adding/removing to the front of a LinkedList vs. an ArrayList. This will help me know how to choose data structures in future programming projects.